APPENDIX I

Web/Internet Applications User Interface Format and Style Guide

Appendix I. Web/Internet Applications User Interface Format and Style Guide

Table of Contents

I.1 INTRODUCTION

This document was developed as part of the Project EASI/ED System-Wide Design Standards Document. The System-Wide Design Standards document defines standards to govern the development of the Project EASI/ED user interface. This will enable Project EASI/ED as a whole to present a consistent user interface to its user community.

I.1.1 Purpose

The purpose of this document is to provide guidance to developers in utilizing the documented standards defined in the Project EASI/ED System-Wide Design Standards Document for user interface design for Web-based applications (as opposed to purely informational Web sites). These standards are intended to address the "look and feel" of the user interface, not how it should function, which is application-specific. However, particularly with respect to Web-based applications, this document provides guidance with respect to standards concerning techniques used in constructing the interface. Given the scenario of multiple contractors being assigned responsibility for specific systems and/or functional areas of Project EASI/ED as a whole, these standards will assure uniformity in the work products that are received by the Department.

The audience to which this document is directed includes those personnel within the Department of Education who will be managing the effort to transform the existing Title IV systems, and particularly the developers that will actually be performing the analysis, design, and construction activities. Consequently, at certain points in the document the content is of necessity quite detailed and technical in character.

I.1.2 Organization of This Document

Section one of this document presents the introduction, which includes descriptions of the purpose and the organization of this document.

Section two presents an overview of the user interface model and navigation strategy that have been developed as the basis of these standards.

Section three examines the EASI/ED prototype and how it demonstrates the standards presented in subsection 6.4.1.1 of the Project EASI/ED System-Wide Design Document.

Section 4 presents an analysis of specific Web-based user interface construction issues, such as screen resolution and the use of graphics.

I.2 USER INTERFACE MODEL AND NAVIGATION STRATEGY

The following subsections present the conceptual model for the standard user interface design recommended for Project EASI/ED Web-based applications, and the strategy devised to define the methods for navigation that are supported by the user interface model.

I.2.1 User Interface Model

The standards that support user interface (UI) capabilities in the Web environment are rapidly evolving. Unfortunately, support for those standards is inconsistent and incomplete in Web browser products from major vendors such as Netscape and Microsoft. This severely constrains the functionality that can be safely provided in a UI if a design objective is to maximize the target audience and minimize the "barriers to entry" experienced by that audience.

In light of this, the user interface model devised for Project EASI/ED Web/Internet applications draws upon both the heritage of the Web itself as primarily a medium for providing access to ***documents***, and of GUI design for client/server systems. At the same time it implicitly recognizes the constraints imposed by the current "aggressively heterogeneous" environment of the Web, where vendors are deliberately deviating from standards in an attempt to distinguish their products, captivate users, and gain competitive advantage. Figure I-1, Project EASI/ED User Interface Model, presents an illustration of the following concepts:

- Content is arranged left-to-right and top-to-bottom, beginning with the organization's logo being placed in the upper left hand corner and the organization's name across the very top of the page to the right of the logo.
- Content is divided into a set of broad functional categories, which are associated with menu options across the top of the page below the organization's name. This arrangement of main menu options is consistent with standards for menu-driven interfaces in computer software such as Microsoft Windows Interface Guidelines for Software Design.
- Content within each broad functional category is divided into topics, subtopics, and (where occasionally necessary) sub-subtopics. Once one of the "main menu" options across the top of the page has been selected, the next page that is presented will have the topics associated with that main menu option displayed vertically on the left hand side of the page as a set of menu options. A brief description of the topics will occupy the "content" section of the page.
- Once one of the "topic menu" options on the left hand side of the page has been selected, the next page that is presented will have the topic name as a static (non-link) title on the left-hand side of the page in an enlarged font size (+1), and below that will be the associated subtopics displayed vertically on the left hand side of the page as a set of menu options.
- Should it be necessary to decompose subtopics into sub-subtopics, the same arrangement as for subtopics will be employed, with the following differences: the topic name becomes an active link and the font will be larger (+2 or +3), and the subtopic name will be the static title below it.

In certain instances—specifically when an individual page is a form, and particularly when it is part of a series of pages that implement a "wizard"-style multi-part form—it may be desirable to exclude menu options from a page in order to make the series of pages as "modal" as possible and strictly minimize a user's ability to deviate from the prescribed pathway through the process. Even then, it is not possible to disable the “Back” button on the browser unless the form is launched in a separate browser instance that does not show the browser menu or toolbar buttons by definition.

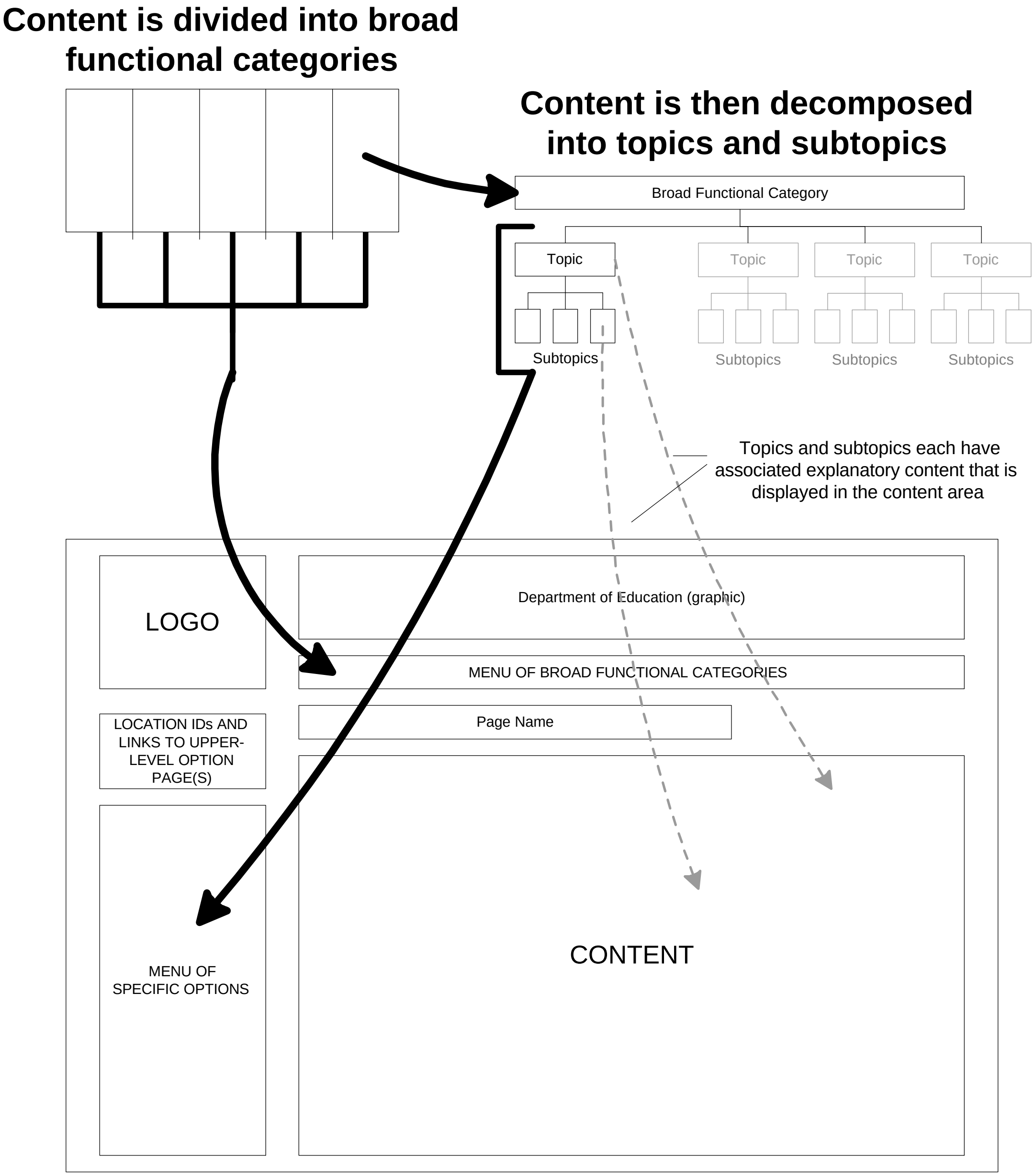

Content is divided into broad functional categories
Content is then decomposed into topics and subtopics
Broad Functional Category
Topic
Topic
Topic
Topic
Subtopics
Subtopics
Subtopics
Subtopics
Topics and subtopics each have associated explanatory content that is displayed in the content area
LOGO
Department of Education (graphic)
MENU OF BROAD FUNCTIONAL CATEGORIES
LOCATION IDs AND LINKS TO UPPER-LEVEL OPTION PAGE(S)
Page Name
MENU OF SPECIFIC OPTIONS
CONTENT

Figure I-1: Project EASI/ED User Interface Model

I.2.2 Navigation Strategy

The navigation strategy is derived directly from the user interface model defined in the preceding section. By organizing content into broad functional categories and then successively decomposing each category into topics, subtopics, and (where absolutely necessary) sub-subtopics, an elegant and easily navigable hierarchy of information can be constructed. The navigation strategy addresses navigation to the site from another site; navigation to other sites from within the site; navigation between pages within the site; and navigation within a single page.

Initial Navigation from Outside Points

Navigation begins with the Universal Resource Locator (URL), which is simply an alias for the rather cryptic Internet Protocol (IP) numeric address. The URL needs to be clear, concise and distinctive so that the user can remember it and easily type it, such as **www.ed.gov**. If possible, the use of subdomain names in the URL (prefixes to the primary domain name, such as **www.easi.ed.gov**) should be restricted to one level of subdomain.

Navigation to Other Web Sites

External hyperlinks need to be in a designated area or on a specified link bar as opposed to being intermingled with hyperlinks that lead to content within the site. Generally developers should refrain from overriding the browser's own defaults for formatting of hyperlinks. In certain situations it may be advisable to provide a "text only" link. However, it is preferable to design the site so as to be usable by text-only browsers such as Lynx in order to avoid the extra work of having to maintain two sites. Note that the ED World Wide Web Policy and Procedures require that a notice be given to users when they are exiting a ".gov" domain and linking to a ".com" domain.

Navigation within the Same Web Site

The user interface model defined in the previous section has the following specific features to support navigation between pages within the web site:

- "Home" button. The logo in the upper left-hand corner of the page should be a link to the "main" page of the site (usually, but not always, "index.html"). This link should be enabled on every page but the main page itself.
- Menu bar. This provides links to all the broad functional categories defined for the content of the web site. It is always at the top of the page. In pages longer than a single screen, a "return to top" link should be provided, or text versions of the links anchored by the main menu should be provided at the bottom of the page.
- Topic menu. This section, arranged vertically along the left-hand side of the page, presents the topics and subtopics associated with a main menu option that has been selected.
- Content area. This defined space where content is presented is bounded by the main menu on the top and the topic menu on the left.

Secondary Menu

In addition to the main menu and topic menus, a secondary menu that provides access to supporting functionality can be provided. This is particularly appropriate for the first (i.e., the “home” or “splash”) page of the site. Links accessed from such a secondary menu could include:

- Site Map
- Search
- Index of topics
- Directory of relevant sites
- Help
- Frequently Asked Questions (FAQ)
- Contacts
- Disclaimers/Legal Notices

Where possible, the site map should be dynamic (i.e., each reference to an item should also be a link to the actual page displaying the item). This is also true of the Index, Contacts and Directory pages, so that each reference to an item or external site should actually be a link. In the case of the Contacts page, this would be a “mailto” link that allows the user to send the contact person an email message if the user’s browser is set up to support email.

Color Coding

An additional feature that extends the user interface model and reinforces the navigation strategy is the use of color-coding. Options on the main menu can be individually color coded, and the topics and subtopics under a specific main menu option would then be presented in the same color as the main menu option they are associated with. This immediately lets users know “where they are” within the site.

Navigation within a Web Page

Every attempt should be made to break content into “chunks” that fit on a single screen so the user can view all content without scrolling down. If content must of necessity extend beyond a single screen, an opening paragraph should describe the content, and provide internal hyperlinks to sections within the page. As stated above, a “return to top” link should be provided at the end of every section accessed by an internal link and at the bottom of the page. Text versions of the links anchored by the main menu may also be provided at the bottom of the page.

I.3 EASI/ED USER INTERFACE PROTOTYPE

The following subsections discuss the various aspects of the EASI/ED User Interface Prototype.

I.3.1 Structure of the EASI/ED Prototype

The EASI/ED prototype was developed using the ColdFusion Application Server to provide database access. The ColdFusion Application Server uses a proprietary tag language, ColdFusion Markup Language (CFML). CFML is used to construct ColdFusion templates (.CFM files) that process database requests and dynamically generate HTML pages. There is only one pure Hypertext Markup Language (HTML) file in the prototype, which is the **index.html** file. This file is blank and an HTTP refresh command to automatically load the **default.cfm** file, which is the first ColdFusion template.

Figure I-2, Structure of Project EASI/ED prototype, illustrates the flow of control through the prototype. The top row represents the main menu options, whereas the subsidiary items are the topic-level menu options. Note that the prototype menu structure does not go below this one level of topic menu options.

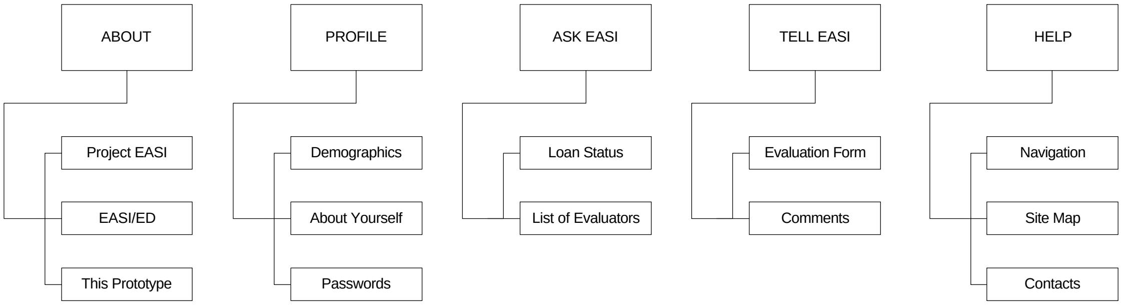
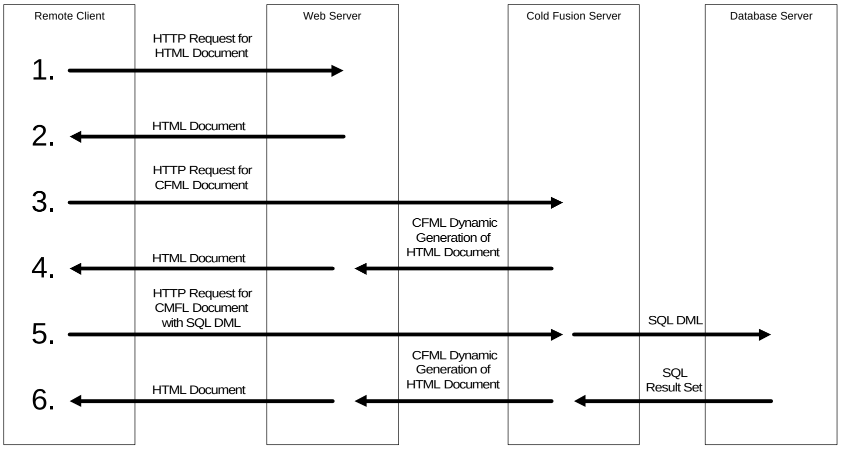


Figure I-2: Structure of Project EASI/ED Prototype



Remote Client
Web Server
Cold Fusion Server
Database Server
1.
HTTP Request for HTML Document
2.
HTML Document
3.
HTTP Request for CFML Document
4.
CFML Dynamic Generation of HTML Document
HTML Document
5.
HTTP Request for CMFL Document with SQL DML
SQL DML
6.
CFML Dynamic Generation of HTML Document
SQL Result Set
HTML Document

Figure I-3: Operation of Prototype Architecture

The process, by which ColdFusion generates dynamic HTML pages, whether or not they require database access, is shown in Figure I-3, Operation of Prototype Architecture. Essentially, developers can embed conditional logic within the ColdFusion templates to determine what HTML should be generated based on various criteria. This is typical of how application servers are used in such architectures, no matter what the particular programming language is that is used to develop the application.

I.3.2 Screen Layout

As shown in Figure I-1, there is a standardized page layout that is derived from the user interface model for Project EASI/ED. Figure I-4, Project EASI/ED Standard Page Layout, illustrates this.

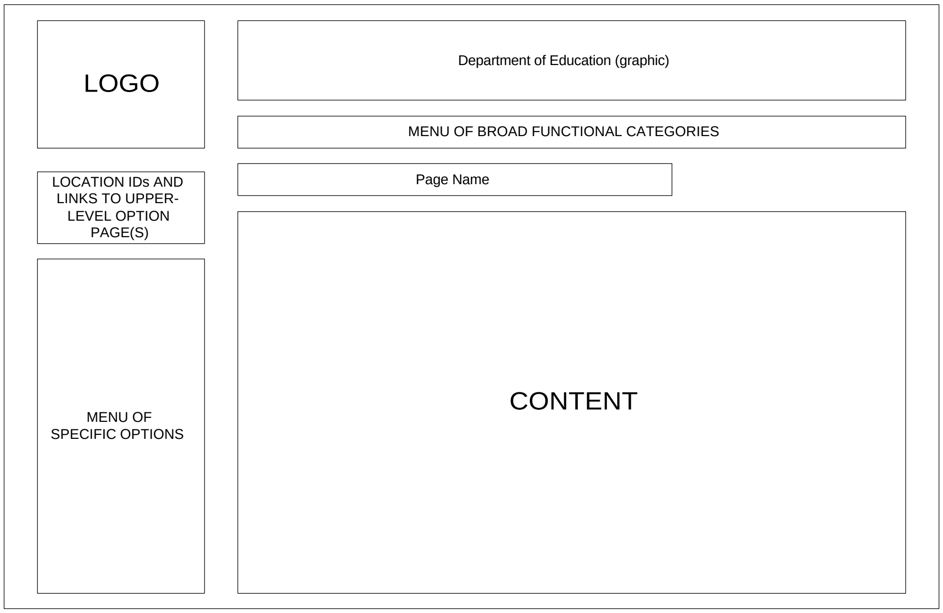
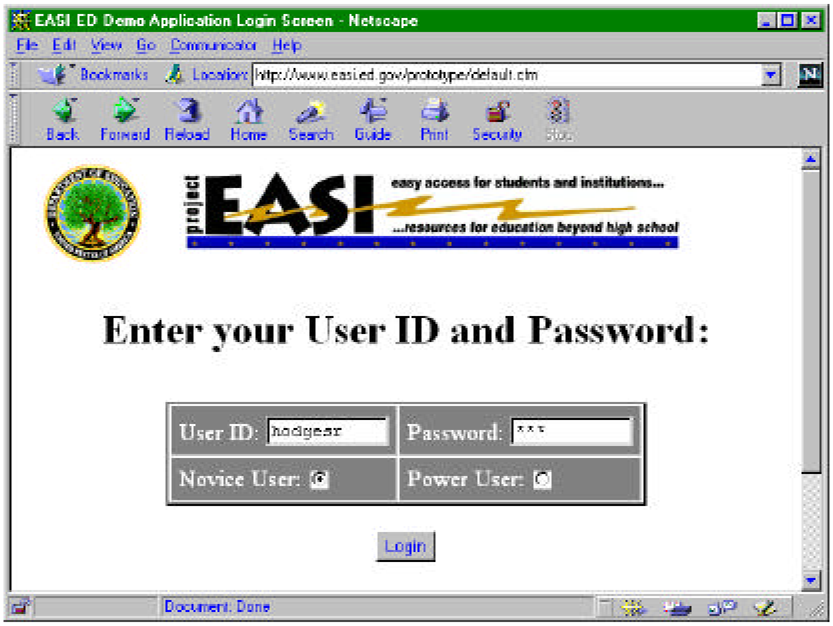

LOGO
Department of Education (graphic)
MENU OF BROAD FUNCTIONAL CATEGORIES
LOCATION IDs AND LINKS TO UPPER-LEVEL OPTION PAGE(S)
Page Name
MENU OF SPECIFIC OPTIONS
CONTENT

Figure 4: Project EASI/ED Standard Page Layout

Figure I-5, Project EASI/ED Prototype Login Screen, shows how the layout has been implemented in a modal login screen.


EASI ED Demo Application Login Screen - Netscape
File Edit View Go Communicator Help
Bookmarks
Location:
Back Forward Reload Home Search Guide Print Security Stop
project EASI
easy access for students and institutions...
...resources for education beyond high school
Login
Document: Done

Location: http://www.easi.ed.gov/prototype/default.cfm

Enter your User ID and Password:

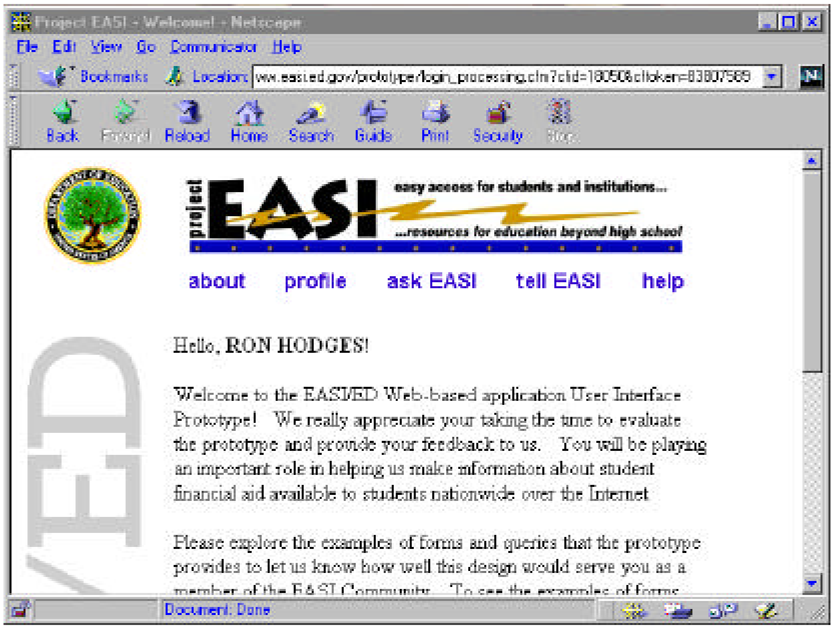
User ID: hodgesr	Password: ***
Novice User: ●	Power User: ○

Figure I-5: Project EASI/ED Prototype Login Screen

Note that no menu options have been incorporated into the login page. However, the user may select between logging in as a Novice User or a Power User. The difference is in how the data entry forms are presented. For this prototype, the User ID and password are stored in the database and the values supplied by the user are authenticated against the stored values. Note that the password is obscured by asterisks.

Figure I-6, Project EASI/ED Prototype Welcome Screen, shows the main screen of the application once the user has been successfully authenticated. Compare the layout of this page to the schematic in figure 4 above. Note that the main screen does not have any menu options on the left hand side of the page—this is because no selections have been made from the main menu that runs across the top of the page underneath the EASI banner. Instead, a graphic that says "PROJECT EASI/ED" runs along the left-hand side of the page as a placeholder. Note also how the white background and the space at the left margin make the page very easy to read.

Note that there is also no secondary menu as described in subsection 2.2 of this document. This is because no mechanism is in place to support such things as a search facility, so instead of providing spurious links that lead to pages with an "under construction" message, the menu options simply are not presented.


Project EASI - Welcome! - Netscape
File
Edit
View
Go
Communicator
Help
Bookmarks
Location:
ww.easi.ed.gov/prototype/login_processing.cfm?cfid=18050&cftoken=83807589
Back
Forward
Reload
Home
Search
Guide
Print
Security
Stop
project EASI
easy access for students and institutions...
...resources for education beyond high school
about
profile
ask EASI
tell EASI
help
ED
Document: Done

Hello, RON HODGES!

Welcome to the EASI/ED Web-based application User Interface Prototype!   We really appreciate your taking the time to evaluate the prototype and provide your feedback to us.   You will be playing an important role in helping us make information about student financial aid available to students nationwide over the Internet

Please explore the examples of forms and queries that the prototype provides to let us know how well this design would serve you as a member of the EASI Community.   To see the examples of forms

Figure I-6: Project EASI/ED Prototype Welcome Screen

Note that the URL has **cfid=18050&cftoken=83807589** appended to the URL. This is an artifact of the ColdFusion application server's mechanism for maintaining state in the web environment. It can use either cookies or append the keys identifying an individual session to the URL.

The prototype is coded to test whether cookies will be accepted, use cookies if they are accepted, or resort to appending the session keys if cookies are not accepted. Cookies are in fact accepted by the browser used to obtain these screen captures, but because a cookie cannot be set and evaluated within the same page, the session keys must be passed to the welcome page by the login page in case the attempt by the login page to set a cookie failed.

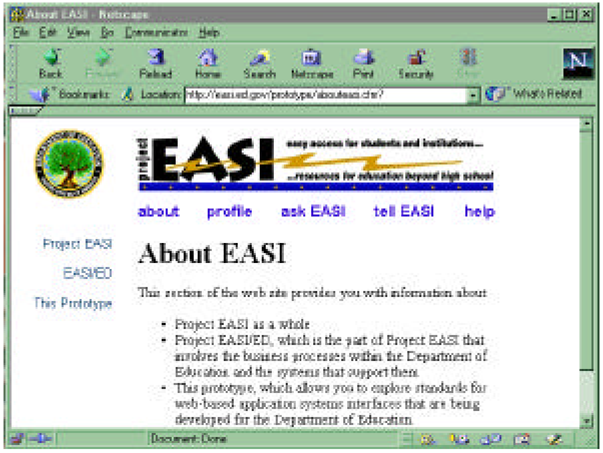
The main menu consists of five options:

- About
- Profile
- Ask EASI
- Tell EASI

- Help

These menu options are highlighted as the user's mouse passes over them, using Javascript functions to switch between pairs of menu option graphics. Such "rollovers" are one of the few permissible uses of Javascript within the standards defined for Web-based applications in subsection 6.4.1.1 of the Project EASI/ED System-Wide Design Standards Document. This is because, where a user's browser does not support it, such functionality will degrade smoothly and not have any adverse impact on the ability of the user to perform the tasks the interface is intended to facilitate.

Figure I-7, Project EASI/ED Prototype About Screen, shows a typical text-based page within the prototype. Note that the menu options on the left-hand side of the page are right justified. These also are highlighted with Javascript rollovers when a user's mouse passes over them. Figure I-8, Project EASI/ED Prototype About Project EASI Screen, shows one of the detail pages presented once a user has selected one of the topic menu options.

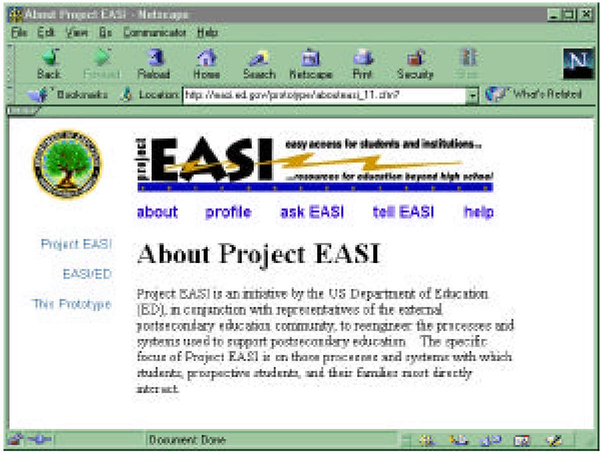

About EASI - Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Stop
Bookmarks Location: http://easi.ed.gov/prototype/abouteasi.cfm?
What's Related
project EASI
easy access for students and institutions...
...resources for education beyond high school!
about
profile
ask EASI
tell EASI
help
Project EASI
EASI/ED
This Prototype
Document: Done

About EASI

This section of the web site provides you with information about

- Project EASI as a whole
- Project EASI/ED, which is the part of Project EASI that involves the business processes within the Department of Education and the systems that support them
- This prototype, which allows you to explore standards for web-based application systems interfaces that are being developed for the Department of Education

Figure I-7: Project EASI/ED Prototype About Screen


About Project EASI - Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Stop
Bookmarks Location: http://easi.ed.gov/prototype/abouteasi_11.cfm?
What's Related
project EASI
easy access for students and institutions...
...resources for education beyond high school!
about
profile
ask EASI
tell EASI
help
Project EASI
EASI/ED
This Prototype
Document: Done

About Project EASI

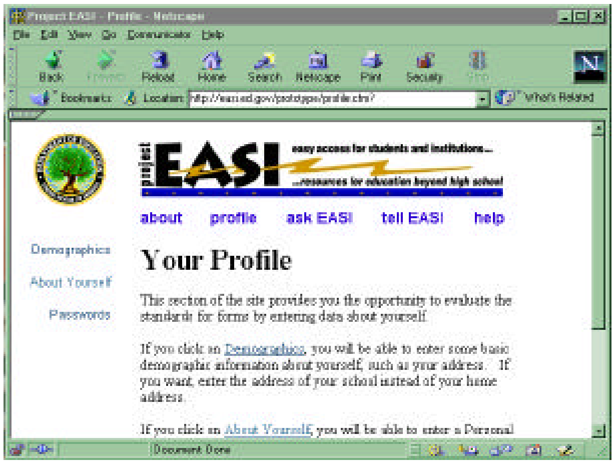
Project EASI is an initiative by the US Department of Education (ED), in conjunction with representatives of the external postsecondary education community, to reengineer the processes and systems used to support postsecondary education. The specific focus of Project EASI is on those processes and systems with which students, prospective students, and their families most directly interact.

Figure I-8: Project EASI/ED Prototype About Project EASI Screen

If the About Project EASI page had its own set of subtopic menus, and one of those was selected, on the subtopic page that was presented it would be appropriate to place a link back to the About Project EASI page above any menu options on that page. The menu options on the left-hand side of the page should be structured in such a way that the user can retrace his or her path without using the "BACK" button on the browser itself.

Figure I-9, Project EASI/ED Prototype Profile Screen, shows the page that presents information about building a user profile in the prototype. The user profile was included as way to demonstrate how forms would appear and operate according to the standards presented in subsection 6.4.1.1 of the Project EASI/ED System-Wide Design Document. Figure I-10, Project EASI/ED Prototype Demographics Screen, shows how the data entry form appears.

As you can see from looking at figure 10, the data entry form is a bit longer than a single screen, but not so much so that it becomes cumbersome or intimidating for a novice user. Compare the size of the screen in figure 10 to that of figure 9, where the browser window is sized to 640 pixels wide by 480 pixels high, which is the size of the screen when the resolution is set to standard VGA. The browser window in figure 10 is actually 664 pixels high, which was possible only because the computer on which the screen capture was performed was set at the XGA resolution of 1024 by 768.


Project EASI - Profile - Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Stop
Bookmarks Location: http://easi.ed.gov/prototype/profile.cfm?
What's Related
project EASI easy access for students and institutions... ...resources for education beyond high school
about profile ask EASI tell EASI help
Demographics
About Yourself
Passwords
Document: Done

Your Profile

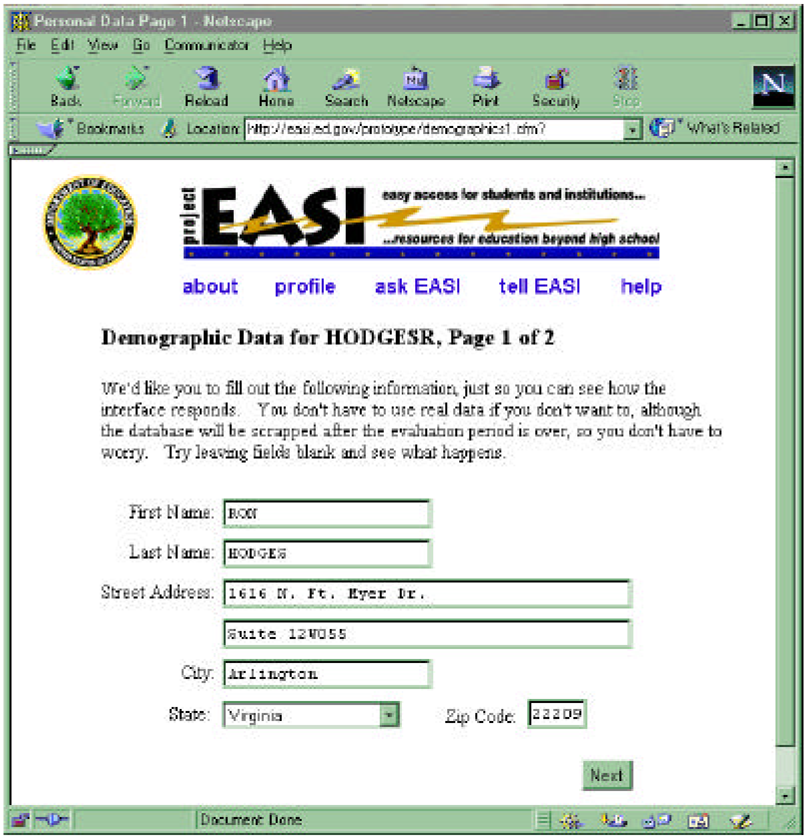
This section of the site provides you the opportunity to evaluate the standards for forms by entering data about yourself.

If you click on Demographics, you will be able to enter some basic demographic information about yourself, such as your address. If you want, enter the address of your school instead of your home address.

If you click on About Yourself, you will be able to enter a Personal

Figure I-7: Project EASI/ED Prototype Profile Screen

If the user had logged in as a “Power User”, both demographics screens would have been combined into a single, longer data entry form to facilitate rapid data entry by a knowledgeable user. One important aspect of both forms is that the button that submits the data for each form is at the bottom of the page, so that a user is unlikely to submit the form inadvertently before it has been completely filled out.


Personal Data Page 1 - Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Stop
Bookmarks Location: http://easi.ed.gov/prototype/demographics1.cfm?
What's Related
project EASI easy access for students and institutions... ...resources for education beyond high school
about profile ask EASI tell EASI help
Next
Document: Done

Demographic Data for HODGESR, Page 1 of 2

We'd like you to fill out the following information, just so you can see how the interface responds. You don't have to use real data if you don't want to, although the database will be scrapped after the evaluation period is over, so you don't have to worry. Try leaving fields blank and see what happens.

First Name: RON
Last Name: HODGES
Street Address: 1616 N. Ft. Myer Dr.
Suite 12W055
City: Arlington
State: Virginia
Zip Code: 22209

Figure I-8: Project EASI/ED Prototype Demographics

It should also be noted that the forms are modeled after “wizards”, which are a series of dialog boxes that walk a user through a process step by step. Thus, each screen has a substantial amount of explanatory text that attempts to proactively answer the most common questions before they have arisen in the user’s mind.

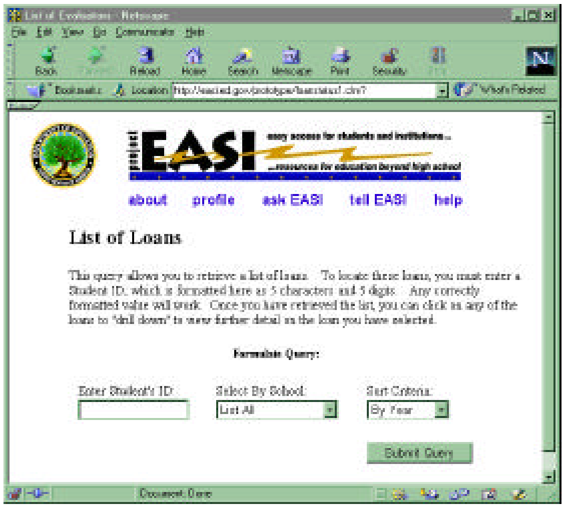
Another key design point is that the data in each section of the form is saved to the database as the user moves through the form, so that if the connection is broken at any point, the data the user has previously entered is not lost and the user can pick up again at the point where he or she left off.

All data validation is conducted on the application server. This is because using Javascript to validate data on the client side can run into numerous problems. The foremost of these is that many browsers do not support Javascript, and on the ones that do, users have the option of turning it off.

Note also that a pulldown list is used for the state name. Actually, the 2-character state abbreviation is what is stored in the database. This eliminates the need for several different kinds of data validation checks on the state code field. This particular GUI control also raises an interesting and challenging issue with respect to coding in HTML versus coding in a conventional client/server development tool.

With a conventional client/server development tool, such a GUI control is typically an object to which parameters can be passed, making it relatively easy to populate it with a list of states extracted from a table, and still have it display the correct state code for a particular record. In HTML, it is not an object *per se*, but part of <SELECT> statement with a set of <OPTION> statements defining the values that will populate the list, any one of which can be tagged as selected (i.e., the default value or value read from the database). The problem facing a developer is how to populate the list from a lookup table and specify the correct value as selected when that value comes from a different table. Fortunately, application servers like ColdFusion provide two different methods to accomplish this—one is to insert ColdFusion conditional logic within the HTML <SELECT> construct, and the other is to use a specific ColdFusion replacement for the <SELECT> construct that is expressly designed to solve this very problem.

Similar issues confront developers when dealing with queries. Fortunately, again the application server provides an answer. Figure I-11, Project EASI/ED Prototype Query Screen, shows how a typical query might be formulated. Figure I-12, Project EASI/ED Prototype Query Results Screen, shows the result set that is returned.


List of Evaluations - Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Stop
Bookmarks Location: http://easi.ed.gov/prototype/loanstatus1.cfm? What's Related
project EASI easy access for students and institutions... ...resources for education beyond high school
about profile ask EASI tell EASI help
Document: Done

List of Loans

This query allows you to retrieve a list of loans. To locate these loans, you must enter a Student ID, which is formatted here as 5 characters and 5 digits. Any correctly formatted value will work. Once you have retrieved the list, you can click on any of the loans to "drill down" to view further detail on the loan you have selected.

Formulate Query:

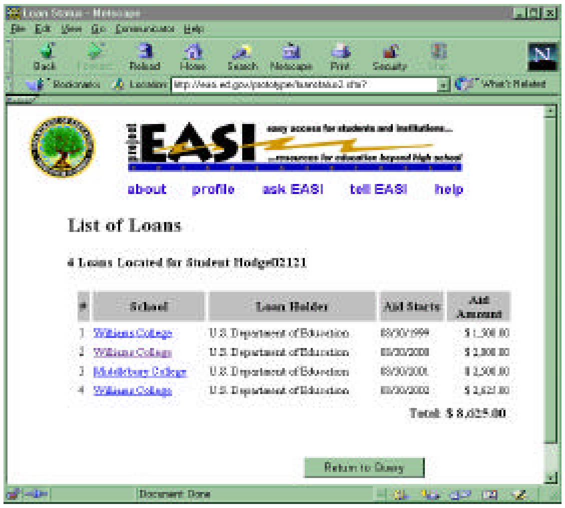
Enter Student's ID:

Select By School: List All

Sort Criteria: By Year

Submit Query

Figure I-9: Project EASI/ED Prototype Query Screen


Loan Status - Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Stop
Bookmarks Location: http://easi.ed.gov/prototype/loanstatus2.cfm? What's Related
project EASI easy access for students and institutions... ...resources for education beyond high school
about profile ask EASI tell EASI help
Document: Done

List of Loans

4 Loans Located for Student Hodge02121

#	School	Loan Holder	Aid Starts	Aid Amount
1	Williams College	U.S. Department of Education	08/30/1999	$ 1,500.00
2	Williams College	U.S. Department of Education	08/30/2000	$ 2,000.00
3	Middlebury College	U.S. Department of Education	08/30/2001	$ 2,500.00
4	Williams College	U.S. Department of Education	08/30/2002	$ 2,625.00
			Total:	$ 8,625.00

Return to Query

Figure I-10: Project EASI/ED Prototype Query Results Screen

Note that the query is parameter-driven, meaning the user supplies values that are inserted into a predefined query that is then executed. Since the user was not identified as a student when he logged in, the form included a field to provide a Student ID number.

The display of the result set not only provides a total for all the values that are displayed; it allows the user to click on the School Name to "drill down" into the data for greater detail. This page could vary in length depending on the number of records returned, but it is a good idea to break the result set into blocks of ten records or less. The primary reason for doing so is that returning large result sets of a few hundred records or more could overwhelm the browser as it attempts to render all the HTML generated by the application server to display the query output.

I.4 User Interface Construction Issues

The following subsections address key aspects with respect to the construction of Web-based user interfaces.

I.4.1 Screen Resolution and Color Depth

One of the aspects that Web-based user interfaces share with user interfaces for conventional client/server systems is that variation in screen resolution and color depth can be problematic. The following discussion provides guidance as to the standard approaches specified for Web-based user interfaces developed for Project EASI/ED.

Screen Resolution

Changing the screen resolution (and especially, changing the system font size from the default of small system fonts to large system fonts) can wreak havoc with even the most beautiful interface design, if it is not constructed to accommodate those changes.

This is because most development tools use absolute screen position addressing, in terms of pixels, inches, centimeters, or “twips”. When screen resolution increases, for example from the VGA standard of 640 by 480 pixels to the XGA standard of 800 by 600 or 1024 by 768, the pixel size becomes smaller, and the proportions of the screen change. If the size of the system font has not been changed from small to large, then the components of the interface merely shrink proportionately as the resolution increases. However, if the size of the system font has changed as well, then the size of the text prompts on the screen will change relative to the size of graphical controls, and the relative positioning of items will be thrown off.

Web developers frequently fall into the trap of designing for their own monitors, which typically are large and support high screen resolutions, which results in HTML pages that are too large to be completely visible at VGA resolution. It is necessary to lay out HTML pages that degrade gracefully as either screen resolution or color depth are reduced down to a specified minimum level, and yet remain readable as the resolution increases.

This minimum standard level of screen resolution for Web applications developed for Project EASI/ED is VGA (640 pixels wide by 480 pixels high—note that the actual width of the browser’s display is approximately 600 pixels or less because of the browser’s own window borders.) Width is the more important consideration, since at 640 x 480 resolution a browser with all tool bars visible presents a relatively small amount of vertical space for the display of content and scrolling may be unavoidable. Graphics should generally be anchored using the upper left-hand corner of the screen as the origin, as opposed to being centered. This does mean that when the browser window is maximized at resolutions above VGA, the interface components will tend to “clump” in the upper left-hand corner of the screen. However, this is preferable to the misalignments that can occur when an attempt is made to align all interface components according to the center of the screen, yet retain their relative positions.

Color Depth

Color depth is defined in terms of the number of bits per pixel used to store color information. The standard color depth scale is as follows:

1 bit	= 2 colors
4 bits	= 16 colors
8 bits	= 256 colors
15 bits	= 32,000 colors
16 bits	= 64,000 colors
24 bits	= 16,000,000,000 colors

The minimum standard color depth for Web applications developed for Project EASI/ED is 8 bits per pixel, which provides a 256-color palette. In reality, however, an evenly distributed palette that has 256 or fewer colors, and which is based on whole number values for the RGB triplets, should contain 216 discrete colors. Thus, there are 40 palette slots not being used, which are at least partially reserved by the operating system (particularly the 20 system colors reserved by Windows). The use of the 216-color "Safety Palette" will ensure that colors display accurately across platforms, and also that image quality will not be degraded by color dithering (dithering occurs when the operating system attempts to replicate a color outside its palette by interweaving two other colors).

I.4.2 Graphics and Page Size

Pages need to be visually interesting, yet they should download quickly. Human factors research has shown that for most computing tasks the threshold of frustration is around 10 seconds. A good rule of thumb is that even a splash page should not be larger than 50 KB in size, including both graphics and text. With careful attention to file format and color usage, a lot can be done with 45 KB or less. File sizes can be a little larger for Intranets, but it is generally not necessary. Reusing the same image on different pages is another way to help solve the bandwidth problem since images are stored in a cache on the user's computer, avoiding multiple downloads.

Properly implemented graphics help retain the audience's attention and highlight or clarify the content of the pages. However, poorly designed graphics can distract the user or make the site difficult to read. Graphics that have too large a file size or that are larger than the user's display will drive them from the site. Also, users should not automatically assume that all graphics would be displayed. Due to memory, processing or bandwidth limitations, some graphics may not be displayed. Developers should identify each graphic with an ALT tag to convey useful information. The IMG SRC tag should specify the height and width of the graphic in pixels to speed the browser's rendering of the graphic.

As stated in the previous section, the minimum standard client configuration is assumed to be 640x480 resolution and 8-bit color depth (256 colors). Developers should use GIF format for line art and non-photographic images. JPG files can be used for 256-color graphics or images based on photography. Because of their higher compression, JPGs are frequently smaller than GIF files. Developers should be aware, however, that JPGs use "lossy compression," which means that each time they are saved, additional detail is shed and the image quality further degrades. The original image should be created and maintained in another format and converted to JPG when it is ready for use.

Animations should be used sparingly, if at all. The audience's attention should be on the content rather than on some gyrations on the screen, particularly if the image is not indicative of the content.

As in the case of good coding style, white space should be used to facilitate comprehension of text. White space and graphics should be used to break up text into easily digestible sections. If text is too crowded and/or cluttered with links and graphical objects, it can be difficult to read at best and baffling to the user at worst. Note the use of white space in the screen captures of the Project EASI/ED prototype that are presented in subsection 3.2.

As stated earlier, it is critically important to recognize that this document specifies standards for user interface (UI) design for Web-based application systems, as opposed to standards for layouts of simple "static" Web sites. The difference is that with the former, a user is attempting to purposefully accomplish something, whereas with the latter the user is essentially a passive observer (much as if the user were watching television). Thus, the primary goal of a user interface for a Web-based application is to help the user accomplish a task, not capture the user's attention with eye-catching design features.

I.4.3 Use of Color

Color should impart information. Developers should use colors that are informative (e.g., red for warnings) wherever possible, while accounting for visual impairment—the text should contain a warning, rather than merely have the warning indicated by the color of the text.

Developers should use colors that are easy on the eyes. In general, bright colors should be used sparingly and only to draw attention to important points or features. A white background is recommended because the contrast focuses the eye on the foreground, specifically on the content. Pages that the user is likely to print should be readable when printed on a black and white printer—white text on a black background should be avoided, for example.

As stated earlier, color coding of sections of the web site can be an aid to navigation. If each vertical section (moving from general to specific in a single subject area) uses the same color coding, and different vertical sections use different color coding, this helps users remember where in the web site they are located. This type of color-coding is easily achieved while still maintaining a visually consistent design. For example, the background color of the navigation bar may change, while the rest of the layout remains consistent. It is important to not change the color scheme of each page dramatically from its predecessor, or the user's ability to build a continuous, comprehensive mental image of the site will be disrupted.